Configure packet forwarding for Kubernetes pods

Published: 2024-04-01

By default, all traffic between nodes in a Kubernetes cluster is seen by the ExtraHop system, because ExtraHop observes all traffic between devices on the wire. Most ExtraHop security detections can be generated from node-level traffic monitoring; however, if you want to monitor traffic between Kubernetes pods for added visibility, you must enable packet forwarding in your Kubernetes cluster. This guide shows you how to deploy a DaemonSet service that configures packet forwarding for each pod in your cluster with the rpcapd software tap.

In addition to configuring packet forwarding, the DaemonSet also deduplicates packets that would otherwise be forwarded multiple times to the ExtraHop sensor.

Before you begin

- Your Kubernetes control plane must be configured on a Linux machine.

Retrieve subnets for Kubernetes pods and services

Before you can configure ExtraHop to monitor Kubernetes pods, you must retrieve the subnets that are allocated to those pods and to the Kubernetes services the pods support.

Important: Note the subnets you retrieve; you will need the subnets in the deployment procedure.

1. Retrieve the subnets for Kubernetes pods.

   The Container Network Interface (CNI) determines which subnets are allocated for Kubernetes pods. The CNI is usually managed by a third-party Kubernetes plug-in. However, if you have not deployed a CNI plug-in, you can retrieve the pod subnet for most Kubernetes deployments by running the following command:

   ```
   kubectl cluster-info dump | grep -m 1 cluster-cidr
   ```

   If you have installed a CNI plug-in, the procedure depends on your CNI provider. For example, with the Calico plug-in, you can retrieve the pod subnet by running the following command:

   ```
   kubectl --namespace=kube-system get daemonset calico-node
    -o=jsonpath='{.spec.template.spec.containers[*].env[?
   (@.name=="CALICO_IPV4POOL_CIDR")].value}
   ```

   For more information about retrieving the pod subnet, see your CNI provider documentation.
2. Retrieve the subnets for your Kubernetes services.

   If you are running a self-managed Kubernetes cluster, you can retrieve the subnet allocated for your services by running the following command:

   ```
   kubectl cluster-info dump | grep -m 1 service-cluster-ip-range
   ```

   If you are running a cloud-managed Kubernetes cluster, the procedure will depend on your cloud provider. For more information, see your cloud provider documentation.

Configure the ExtraHop system to discover pods

With L2 discovery, the ExtraHop system assigns all IP addresses to an associated L2 device; this is the default setting for ExtraHop systems. If L2 discovery is enabled, you must configure the ExtraHop system

to discover Kubernetes pods as remote devices, even if the pods are located on nodes inside your local network. Otherwise, the pod IP addresses will only be associated with the corresponding L2 devices for the Kubernetes nodes, and the system will not track the pods as separate devices.

1. Enable RPCAP on the ExtraHop system.
   a) Configure RPCAP on the ExtraHop system.
   b) Configure a packet-forwarding rule for the pod subnet on the ExtraHop system.
      - Note the port number you select. You will need the number in the deployment procedure.
      - In the Interface Address field, specify the pod subnet as a CIDR block.
      - Leave the Interface Name field blank.
      - Leave the Filter field blank.
   c) Save the running configuration file.
2. Configure the ExtraHop system to discover pods as remote L3 devices.
   In the Remote Device Discovery section, specify the pod subnet that you retrieved in the previous procedure.

   Important: This step is required only if L2 discovery is enabled. If you have enabled L3 discovery for local devices, skip this step.

Create the rpcapd container image

Create a container image for the containers that will forward packets to the ExtraHop system.

Note: The following instructions show you how to create the container image with Docker. However, you can create the image with any tool that produces Open Container Initiative (OCI) compliant images.

1. Go to the ExtraHop code-examples GitHub repository and download the `deploy_kubernetes_daemon` directory.
2. Download the RPCAP install files to the `deploy_kubernetes_daemon` directory.
   Click the download link under Installation package for Ubuntu 22.04.
3. Open a terminal application and navigate to the `deploy_kubernetes_daemon` directory.
4. Run the following command to create the docker container image:

   ```
   docker build -t rpcapd --build-arg
    RPCAPD_DEB_ARCHIVE=<RPCAP_install_file> .
   ```

   Replace <RPCAP_install_file> with the filename of the RPCAP install file.
5. Tag the image in a registry that can be accessed by all nodes in your Kubernetes cluster:

   ```
   docker tag rpcapd EXAMPLE-REGISTRY/rpcapd:latest
   ```

   Note: You must replace `EXAMPLE_REGISTRY` with the name of your registry.
6. Push the image to the registry:

   ```
   docker image push EXAMPLE-REGISTRY/rpcapd:latest
   ```

Deploy the rpcapd DaemonSet service

1. Write the DaemonSet spec file.
   a) Go to the ExtraHop code-examples GitHub repository and download the `deploy_kubernetes_daemon/rpcapd_daemon.yaml` file to the `rpcapd` directory.

b) In the `rpcapd` directory, open the `rpcapd_daemon.yaml` file in a text editor.
c) Replace the values for the following variables with information from your environment:

image

The name and registry location of the image you created in the previous procedure. For example:

```
EXAMPLE-REGISTRY/rpcapd:latest
```

env.name = EXTRAHOP_SENSOR_IP

The IP address of the ExtraHop sensor

env.name = RPCAPD_TARGET_PORT

The port on the ExtraHop sensor that you assigned the packet-forwarding rule to.

env.name = PODNET

The subnets of the pods in your cluster that you retrieved earlier, in a comma-separated list.

env.name = SVCNET

The subnets of the services in your cluster that you retrieved earlier, in a comma-separated list.

d) Save and close the `rpcapd_daemon.yaml` file.

2. Deploy the DaemonSet by running the following command:

```
kubectl apply -f rpcapd_daemon.yaml
```

The system displays output similar to the following text:

```
namespace/extrahop created
daemonset.apps/extrahop-rpcapd created
```

3. Confirm that the deployment was successful:

```
kubectl wait pod -n extrahop -l component=extrahop-rpcapd --
for=condition=Ready
```

When a pod is deployed, the command displays output similar to the following text:

```
pod/extrahop-rpcapd-vfctb condition met
```

After every pod is deployed, the command exits.

You can now view metrics for Kubernetes pods in the ExtraHop system.